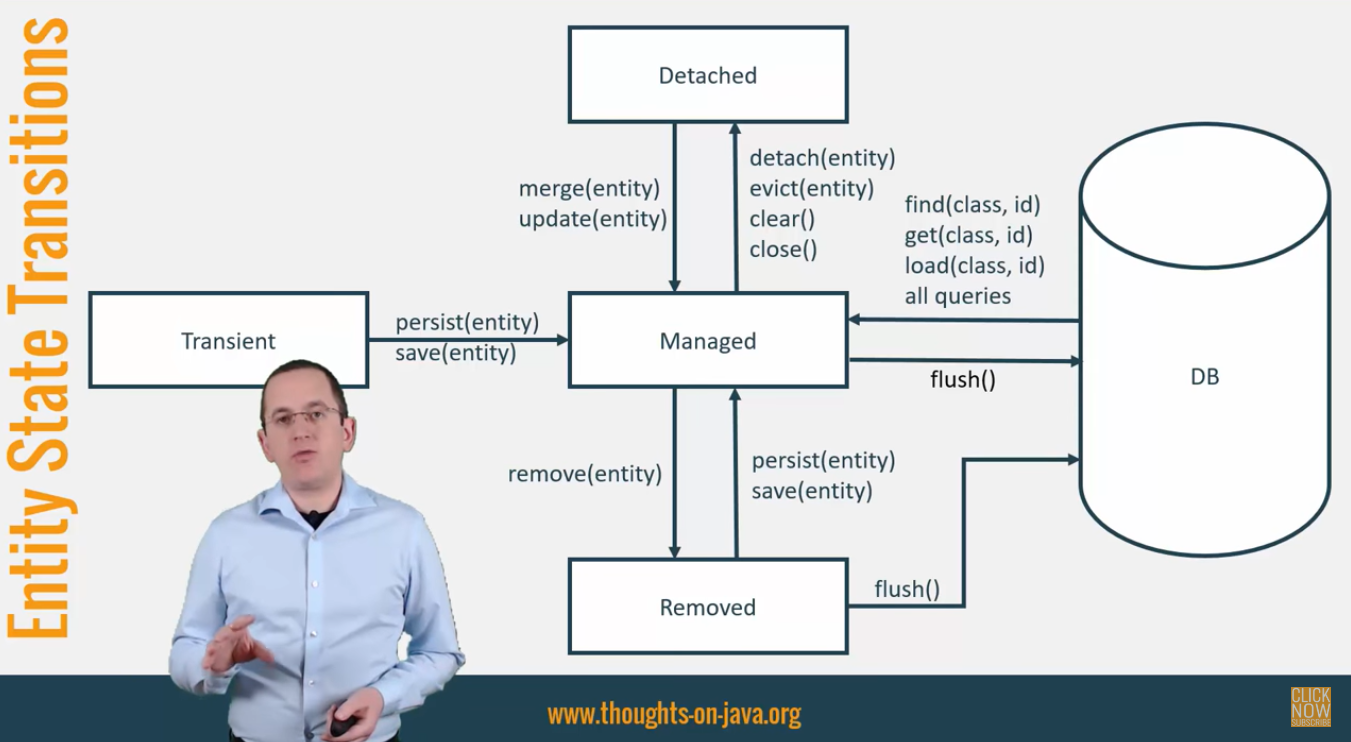


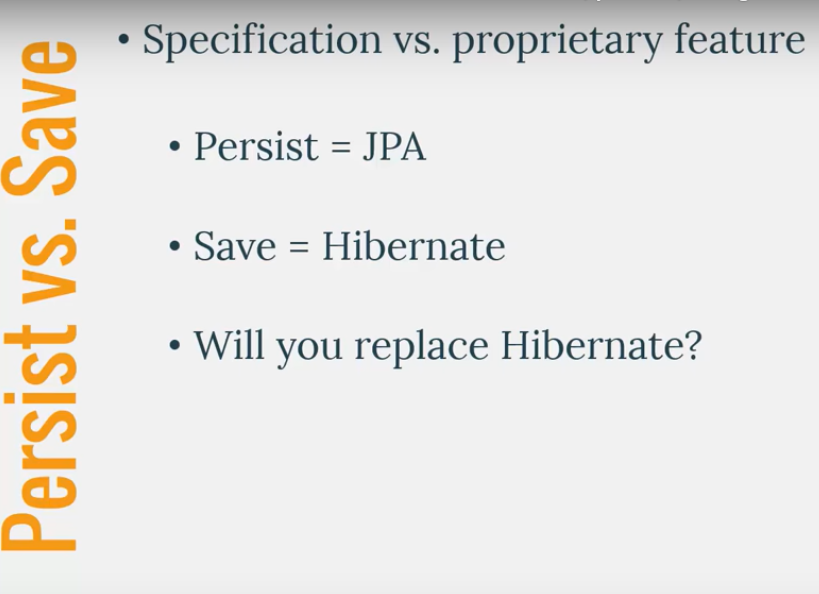
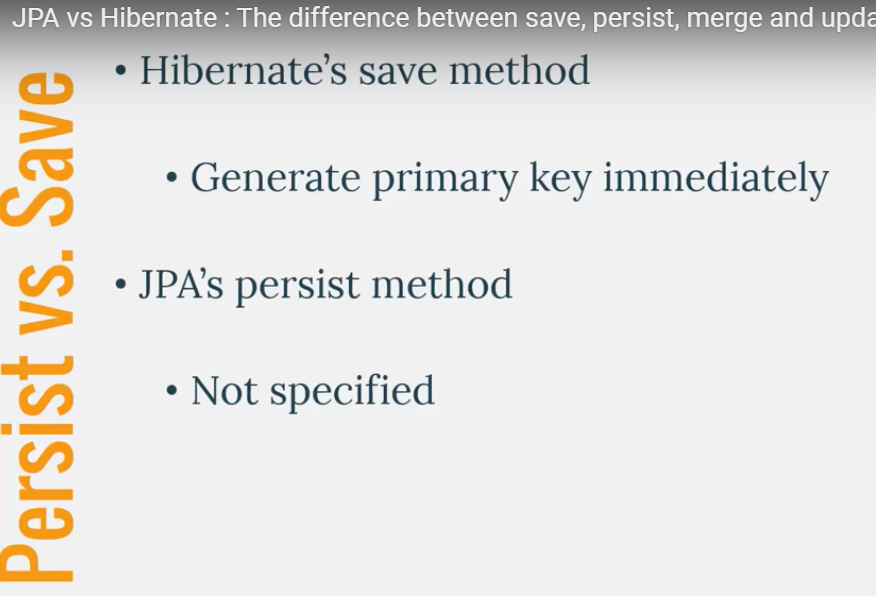
Entity States:

==========



**From Transient to Managed state:**

JPA Persist method or hibernate save method .

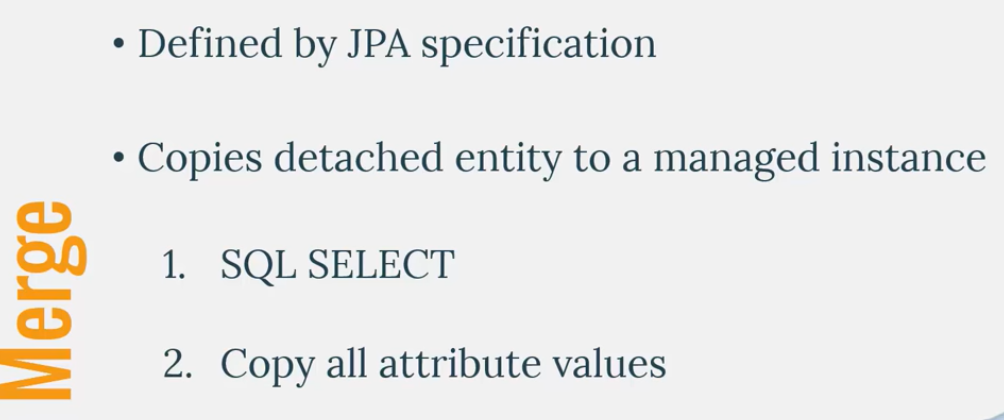
Hibernates delays execution for identity primary key case.

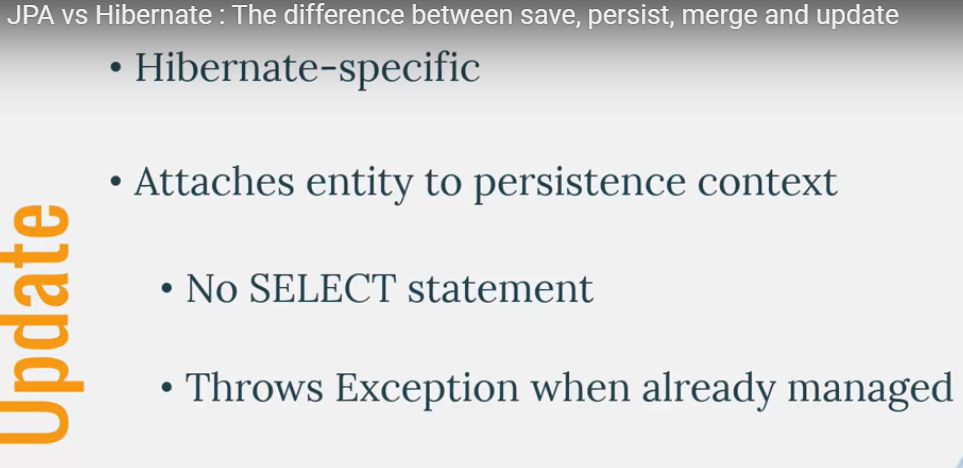
Persist method generate primary during flush or after execution of persist method.

**Update or Merge methods:**

Can be used to when entity is in detach mode meaning entity not exist in first level cache.

Hibernate Update or JPA merge method





No select statement : but you can annotation Entity with @SelectBeforeupdate

Hibernate updates the only provided entity not entity Graph. But merge will update all entityGraph.





There are a number of scopes for defining fetching:

*static*

Static definition of fetching strategies is done in the mappings. The statically-defined fetch strategies is used in the absence of any dynamically defined strategies

SELECT

Performs a separate SQL select to load the data. This can either be EAGER (the second select is issued immediately) or LAZY (the second select is delayed until the data is needed). This is the strategy generally termed N+1.

JOIN

Inherently an EAGER style of fetching. The data to be fetched is obtained through the use of an SQL outer join.

BATCH

Performs a separate SQL select to load a number of related data items using an IN-restriction as part of the SQL WHERE-clause based on a batch size. Again, this can either be EAGER (the second select is issued immediately) or LAZY (the second select is delayed until the data is needed).

SUBSELECT

Performs a separate SQL select to load associated data based on the SQL restriction used to load the owner. Again, this can either be EAGER (the second select is issued immediately) or LAZY (the second select is delayed until the data is needed).

*dynamic* (sometimes referred to as runtime)

Dynamic definition is really use-case centric. There are multiple ways to define dynamic fetching:

*fetch profiles*

defined in mappings, but can be enabled/disabled on the Session.

HQL/JPQL

and both Hibernate and JPA Criteria queries have the ability to specify fetching, specific to said query.

entity graphs

Starting in Hibernate 4.2 (JPA 2.1) this is also an option.

Onetomany : fetch type is lazy,

Manytoone : fetch type is eager